

- [46] Q. Stiévenart and C. De Roover, “Wassail: a webassembly static analysis library,” in *Fifth International Workshop on Programming Technology for the Future Web*, 2021.
- [47] F. Breitfelder, T. Roth, L. Baumgärtner, and M. Mezini, “Wasma: A static webassembly analysis framework for everyone,” in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 753–757, 2023.
- [48] S. Cao, N. He, Y. Guo, and H. Wang, “WASMixer: Binary Obfuscation for WebAssembly,” *arXiv e-prints*, p. arXiv:2308.03123, Aug. 2023.
- [49] W. Fu, R. Lin, and D. Inge, “Taintassembly: Taint-based information flow control tracking for webassembly,” *arXiv preprint arXiv:1802.01050*, 2018.
- [50] Q. Stiévenart, D. Binkley, and C. De Roover, “Dynamic slicing of webassembly binaries,” in *39th IEEE International Conference on Software Maintenance and Evolution*, IEEE, 2023.
- [51] Q. Stiévenart, D. W. Binkley, and C. De Roover, “Static stack-preserving intra-procedural slicing of webassembly binaries,” in *Proceedings of the 44th International Conference on Software Engineering, ICSE ’22*, (New York, NY, USA), p. 2031–2042, Association for Computing Machinery, 2022.
- [52] D. Lehmann and M. Pradel, “Wasabi: A framework for dynamically analyzing webassembly,” in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 1045–1058, 2019.
- [53] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. Tullsen, and D. Stefan, “Swivel: Hardening WebAssembly against spectre,” in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1433–1450, USENIX Association, Aug. 2021.
- [54] M. Kolosick, S. Narayan, E. Johnson, C. Watt, M. LeMay, D. Garg, R. Jhala, and D. Stefan, “Isolation without taxation: Near-zero-cost transitions for webassembly and sfi,” *Proc. ACM Program. Lang.*, vol. 6, jan 2022.
- [55] E. Johnson, E. Laufer, Z. Zhao, D. Gohman, S. Narayan, S. Savage, D. Stefan, and F. Brown, “Wave: a verifiably secure webassembly sandboxing runtime,” in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2940–2955, 2023.
- [56] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, “New kid on the web: A study on the prevalence of webassembly in the wild,” in *Detection of Intrusions and Malware, and Vulnerability Assessment: 16th International Conference, DIMVA 2019, Gothenburg, Sweden, June 19–20, 2019, Proceedings 16*, pp. 23–42, Springer, 2019.

- [57] S. Bhansali, A. Aris, A. Acar, H. Oz, and A. S. Uluagac, “A first look at code obfuscation for webassembly,” in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec ’22, (New York, NY, USA), p. 140–145, Association for Computing Machinery, 2022.
- [58] B. Baudry and M. Monperrus, “The multiple facets of software diversity: Recent developments in year 2000 and beyond,” *ACM Comput. Surv.*, vol. 48, sep 2015.
- [59] K. Pohl, G. Böckle, and F. Van Der Linden, *Software product line engineering: foundations, principles, and techniques*, vol. 1. Springer, 2005.
- [60] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, “Managing performance vs. accuracy trade-offs with loop perforation,” in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ESEC/FSE ’11, (New York, NY, USA), p. 124–134, Association for Computing Machinery, 2011.
- [61] Avizienis and Kelly, “Fault tolerance by design diversity: Concepts and experiments,” *Computer*, vol. 17, no. 8, pp. 67–80, 1984.
- [62] T. Y. Chen, F.-C. Kuo, R. G. Merkel, and T. H. Tse, “Adaptive random testing: The art of test case diversity,” *J. Syst. Softw.*, vol. 83, pp. 60–66, 2010.
- [63] T. Jackson, *On the Design, Implications, and Effects of Implementing Software Diversity for Security*. PhD thesis, University of California, Irvine, 2012.
- [64] G. R. Lundquist, V. Mohan, and K. W. Hamlen, “Searching for software diversity: Attaining artificial diversity through program synthesis,” in *Proceedings of the 2016 New Security Paradigms Workshop*, NSPW ’16, (New York, NY, USA), p. 80–91, Association for Computing Machinery, 2016.
- [65] J. C. Knight and N. G. Leveson, “An experimental evaluation of the assumption of independence in multiversion programming,” *IEEE Trans. Softw. Eng.*, vol. 12, p. 96–109, jan 1986.
- [66] B. Randell, “System structure for software fault tolerance,” *SIGPLAN Not.*, vol. 10, p. 437–449, apr 1975.
- [67] N. Harrand, *Software Diversity for Third-Party Dependencies*. PhD thesis, KTH, Software and Computer systems, SCS, 2022. QCR 20220413.
- [68] J. V. Cleemput, B. Coppens, and B. De Sutter, “Compiler mitigations for time attacks on modern x86 processors,” *ACM Trans. Archit. Code Optim.*, vol. 8, jan 2012.